REMARKS

The Examiner is thanked for his courtesy during a telephonic interview on 27 July 2005. The following remarks include at least a summary of our discussion. Claims 1, 5-8, 17, 18, 20, 22, 24-25, 29-32, and 40-41 have been amended to clarify the invention. Claims 3, 10, 21, 23, 27, and 33 have been cancelled. Claims 1-2, 4-8 and 11-20, 22, 24-26, 28-32, and 34-41 remain pending.

The Examiner rejected claims 1-8 and 10-41 under 35 U.S.C. §103(a) as being unpatentable over Li (U.S. Patent 5,643,125) and further in view of Dreszer (U.S. Patent 6,442,661). The Examiner's rejections are respectfully traversed as follows.

Claim 1 is directed towards a "method for assigning traffic buckets to a cache system." Claim also requires the following steps:

> a) when a new cache system starts up in a cache cluster having a plurality of cache systems among which a plurality of total buckets are to be allocated, determining a full bucket allocation for the new cache system and assigning a portion of the full bucket allocation to the new cache system that was determined at start up;
> b) periodically determining a load of the new cache system;
> c) each time it is periodically determined that the new cache system is underloaded and buckets have not been previously shed from the new cache system, slowly assigning a portion of the full bucket allocation that was determined at startup unless the cache cluster is operating at a maximum load;
> d) each time it is periodically determined that the new cache system is underloaded and buckets have been shed previously from the new cache system, slowly re-assigning a portion of the buckets that have been previously shed from the new cache system to the new cache system unless the cache cluster is operating at a maximum load; and
> e) each time it is periodically determined that the new cache system is overloaded, shedding a portion of the buckets previously assigned to the new cache system,
> wherein each bucket portion corresponds to a portion of the total traffic being handled by the cache cluster.

In other words, when a new cache system starts up, a full bucket allocation is first determined for this new cache and then only a portion of this determined full bucket allocation is assigned to the new cache system. The load of the new cache is then periodically monitored.

When the new cache is overloaded or underloaded, only a portion of either the full bucket allocation that was determined at startup, previously shed buckets, or previously assigned buckets are assigned or shed.

Embodiments of the present invention allow efficient distribution of traffic buckets among a plurality of cache systems of a cache cluster system without quickly overwhelming the cache systems by assigning too much load at one time. At startup, bucket *portions* are *slowly* assigned to the new cache system in a slow startup process. At startup, only a portion of the full bucket allocation is initially assigned to the new cache. Based on periodic monitoring of the new cache system's load, traffic bucket portions of the full allocation are then slowly assigned to this new cache system when buckets have not been previously shed. Portions of the buckets that have been previously shed are assigned when buckets have been shed previously These mechanisms allow the new cache system to not be quickly overwhelmed by too much traffic. By applying this technique to each new cache system as it enters the cache cluster, traffic buckets can be slowly and periodically distributed among the cache systems of the cache cluster while managing the traffic load of individual cache systems so they don't remain overloaded.

Conventionally, traffic buckets were evenly allocated between the available cache systems of a cache cluster and this determined full allocation is immediately assigned to each cache system. For various reasons, a particular cache system may be overloaded with traffic under this simple distribution scheme. For instance, a particular cache system may be unable to handle as much traffic as another cache system. See Background Section, Page 4, Lines 16-22.

The primary reference Li describes a parallel database system for storing tables at multiple nodes. Li also describes techniques for storing tables at a node that is being added to the parallel database system in Col. 5, Line 38 through Col. 6, Line 52. Even if one argues that distribution of tables among nodes is the same as distribution of traffic bucket portions to cache systems, Li fails to teach or suggest mechanisms for slowly assigning or shedding portions of either the full bucket allocation that was determined at startup, previously shed buckets, or previously assigned buckets, in the manner claimed. At startup of a node, Li merely describes redistributing some of the tables from the existing nodes to the new node. See Col. 5, Lines 48-44. That is, the new node receives its full table allocation and not a portion of its full table allocation. Li further describes this redistribution process as including a step of determining what buckets of data (or table data) to move from each existing node to the new node. See step 154 of Fig. 6A and Col. 6, Lines 24-26. The tables of each existing node are then scanned and gathered in a communication buffer, and the buffer is then emptied into the new node. See Col. 6, Lines 31-38. Although Li briefly mentions that "load balancing is the primary goal", this statement is in the context of "[w]hen adding new nodes to the PDB system, the ideal

arrangement is shifting work load from existing nodes to the new nodes to distribute the work load evenly." See Col. 6, Lines 53-56. As described above, this redistribution process appears to be a one time distribution process that is performed at startup of a new node and is not carried out in a slow-start manner. Li also fails to teach or suggest mechanisms for slowly shedding or assigning previously assigned or previously shed bucket *portions*, respectively. In sum, Li fails to teach or suggest mechanisms for slowly assigning or shedding portions of either the full bucket allocation that was determined at startup, previously shed buckets, or previously assigned buckets, in the manner claimed.

The secondary reference Dreszer also suffers from at least some of the above described deficiencies. For example, Dreszer fails to teach or suggest a slow-start mechanisms, in the manner claimed. Specifically, Dreszer fails to teach or suggest two mechanisms for (1) determining a full bucket allocation for the new cache system and (2) then slow assignment of portions of the full bucket allocation to a new cache system, in the manner claimed.

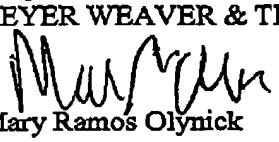
In general, Dreszer is directed towards managing a memory cache of a single system and not assigning buckets to a new cache system that is starting up in a cache system having multiple cache systems among which buckets are allocated, in the manner claimed. Although Dreszer teaches periodically trimming of queue sizes in a single system, Dreszer merely teaches trimming the queue sizes based on each memory request. See Column 12, Lines 21-22: "Fig. 11 shows an example flow diagram of an embodiment of step 70 of FIG. 4 for adjusting the sizes queues 40 in relation to memory request/requirements" and Lines 62-66: "the selected size queue 40 includes a free segment 42 of sufficient size, wherein the segment 42 is allocated to satisfy the request (step 232)." In other words, Dreszer teaches the size queue's segment is allocated based on the request's size requirement and the size queue's segment allocation is not slowly assigned to the size queue in portions, in the manner claimed.

For the forgoing reasons, it is respectfully submitted that claims 1, 18, 25, and 41 are patentable over the cited references.

The Examiner's rejections of the dependent claims are also respectfully traversed. However, to expedite prosecution, all of these claims will not be argued separately. Claims 2-9, 11-17, 19-24, and 26-40 each depend directly from independent claims 1, 18, or 25 and, therefore, are respectfully submitted to be patentable over cited art for at least the reasons set forth above with respect to claims 1, 18, and 25. Further, the dependent claims require additional elements that when considered in context of the claimed inventions further patentably distinguish the invention from the cited art.

Applicant believes that all pending claims are allowable and respectfully requests a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER & THOMAS, LLP



Mary Ramos Olynick
Reg. 42,963

P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100